# An object-oriented representation of speech for EARS:

There are four general object categories to be represented. They are STT objects, MDE objects, source (speaker) objects, and "segments".[1] Each of these general categories may be represented by one or more types and subtypes, as shown in table 1.

**Table 1 Rich Text object types and subtypes**

| Type | Subtypes |
|---|---|
| **SEGMENT** | **eval**, or (none) |
| **STT types:** | |
| **LEXEME** | **lex**, **fp**, **frag**, **un-lex**[2], **for-lex**, **alpha**[3], **acronym**[3], **interjection**[3], **propername**[3], and **other** |
| **NON-LEX** | **laugh**, **breath**, **lipsmack**, **cough**, **sneeze**, and **other** |
| **NON-SPEECH** | **noise**, **music**, and **other** |
| **MDE types:** | |
| **FILLER** | **filled_pause**, **discourse_marker**, **explicit_editing_term**, and **other** |
| **EDIT** | **repetition**, **restart**, **revision**, **simple**, **complex**, and **other** |
| **IP** | **edit**, **filler**, **edit&filler**, and **other** |
| **SU** | **statement**, **backchannel**, **question**, **incomplete**, **unannotated**, and **other** |
| **CB** | **coordinating**, **clausal**, and **other** |
| **A/P** | (none) |
| **SPEAKER** | (none) |
| **Source information:** | |
| **SPKR-INFO** | **adult_male**, **adult_female**, **child**, and **unknown** |

Each of these objects (except for **SEGMENT**) represents an EARS research target. And, except for the static speaker information object [ **SPKR-INFO**], each object exhibits a temporal extent with a beginning time and a duration. (The duration of interruption points [ **IP**] and clausal boundaries [**CB**] is zero by definition.)

These objects are represented individually, one object per record, using a flat record format with object attributes stored in white-space separated fields. The format is shown in table 2.

[1] Although the "segment" is an artificial construct, it is important because it is produced by LDC to provide a modicum of temporal organization in the annotation.
[2] Un-lex is also used to tag words that are infected with or affected by laughter.
[3] This subtype is an optional addition to the previous set of lexeme subtypes which is provided to supplement the interpretation of some lexemes.

**Table 2 Object record format for EARS objects**

| Field 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|
| type | file | chnl | tbeg | tdur | ortho | stype | name | conf |

where

`file` is the waveform file base name (i.e., without path names or extensions).

`chnl` is the waveform channel (e.g., "**1**" or "**2**").

`tbeg` is the beginning time of the object, in seconds, measured from the start time of the file.[4] If there is no beginning time, use `tbeg` = "**<NA>**".

`tdur` is the duration of the object, in seconds.[4] If there is no duration, use `tdur` = "**<NA>**".

`stype` is the subtype of the object. If there is no subtype, use `stype` = "**<NA>**".

`ortho` is the orthographic rendering (spelling) of the object for STT object types. If there is no orthographic representation, use `ortho` = "**<NA>**".

`name` is the name of the speaker. `name` must uniquely specify the speaker within the scope of the `file`. If `name` is not applicable or if no claim is being made as to the identity of the speaker, use `name` = "**<NA>**".

`conf` is the confidence (probability) that the object information is correct. If `conf` is not available, use `conf` = "**<NA>**".

This format, when specialized for the various object types, results in the different field patterns shown in table 3.

**Table 3 Format specialization for specific object types**

| Field 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|
| *type* | *file* | *chnl* | *tbeg* | *tdur* | *ortho* | *stype* | *name* | *conf* |
| **SEGMENT** | `file` | `chnl` | `tbeg` | `tdur` | **<NA>** | **eval** or **<NA>** | `name` or **<NA>** | `conf` or **<NA>** |
| **LEXEME**<br>**NON-LEX** | `file` | `chnl` | `tbeg` | `tdur` | `ortho` or **<NA>** | `stype` | `name` | `conf` or **<NA>** |
| **NON-SPEECH** | `file` | `chnl` | `tbeg` | `tdur` | **<NA>** | `stype` | **<NA>** | `conf` or **<NA>** |
| **FILLER**<br>**EDIT**<br>**SU** | `file` | `chnl` | `tbeg` | `tdur` | **<NA>** | `stype` | `name` | `conf` or **<NA>** |
| **IP**<br>**CB** | `file` | `chnl` | `tbeg` | **<NA>** | **<NA>** | `stype` | `name` | `conf` or **<NA>** |
| **A/P**<br>**SPEAKER** | `file` | `chnl` | `tbeg` | `tdur` | **<NA>** | **<NA>** | `name` | `conf` or **<NA>** |
| **SPKR-INFO** | `file` | `chnl` | **<NA>** | **<NA>** | **<NA>** | `stype` | `name` | `conf` or **<NA>** |

[4] If `tbeg` and `tdur` are "fake" times that serve only to synchronize events in time and that do not represent actual times, then these times should be tagged with a trailing asterisk (e.g., `tbeg` = **12.34*** rather than **12.34**).

## Transforming EARS objects to a sequence of events:

For purposes of training recognition algorithms it may be desirable to transform the object data into a chronological sequence of object boundary events. This may be done by representing separately the beginning and end of every object and then sorting these event records into chronological order. The record format for these temporally sequenced records is identical to the record format for the objects themselves, except for three additional fields used to identify the type of boundary, the object that the boundary derives from, and the boundary time. The format is shown in table 4.

**Table 4 Event record format for temporally sequenced events**

| Field 1 | 2 | 3 | {Fields 4…12} |
|---|---|---|---|
| eventType | objectID | eventTime | {object fields 1…9} |

where

`eventType` is the type of event being represented, namely beg, end, and obj (where obj is an object that is represented by a single point in time). For speaker information, which is not a function of time, use `eventType` = "**<NA>**".

`objectID` is an (arbitrary) ID that is unique to the object that this event derives from. For speaker information, which is not a function of time, use `eventType` = "**<NA>**".

`eventTime` is the time of the event, in seconds, measured from the start time of the file. [5] For speaker information, which is not a function of time, use `eventType` = "**<NA>**".

This format, when specialized for the various event types, results in the different field patterns shown in table 5.

**Table 5 Format specialization for specific event types**

| Field 1 | 2 | 3 | 4 | {Fields 5…12} |
|---|---|---|---|---|
| *eventType* | *objectID* | *eventTime* | *objectType* | *{object fields 2…9}* |
| **beg\|end** | objectID | eventTime | **SEGMENT**<br>**LEXEME**<br>**NON-LEX**<br>**NON-SPEECH**<br>**FILLER**<br>**EDIT**<br>**SU**<br>**A/P**<br>**SPEAKER** | {object fields 2…9} |
| **obj** | objectID | eventTime | **IP**<br>**CB** | {object fields 2…9} |
| **<NA>** | **<NA>** | **<NA>** | **SPKR-INFO** | {object fields 2…9} |

[5] If `eventTime` is a "fake" time that serves only to synchronize events in time and that does not represent an actual time, then this time should be tagged with a trailing asterisk (e.g., `eventTime` = **12.34*** rather than **12.34**).